entropy

MDPI

Article

TIPS: A Framework for Text Summarising with Illustrative Pictures

Justyna Golec †, Tomasz Hachaj *,† and Grzegorz Sokal

Institute of Computer Science, Pedagogical University of Krakow, 2 Podchorazych Ave, 30-084 Krakow, Poland; justyna.golec@up.krakow.pl (J.G.); grzegorz.sokal@up.krakow.pl (G.S.)

* Correspondence: tomekhachaj@o2.pl; Tel.: +48-126-627-845

† These authors contributed equally to this work.

Abstract: We propose an algorithm to generate graphical summarising of longer text passages using a set of illustrative pictures (TIPS). TIPS is an algorithm using a voting process that uses results of individual "weak" algorithms. The proposed method includes a summarising algorithm that generates a digest of the input document. Each sentence of the text summary is used as the input for further processing by the sentence transformer separately. A sentence transformer performs text embedding and a group of CLIP similarity-based algorithms trained on different image embedding finds semantic distances between images in the illustration image database and the input text. A voting process extracts the most matching images to the text. The TIPS algorithm allows the integration of the best (highest scored) results of the different recommendation algorithms by diminishing the influence of images that are a disjointed part of the recommendations of the component algorithms. TIPS returns a set of illustrative images that describe each sentence of the text summary. Three human judges found that the use of TIPS resulted in an increase in matching highly relevant images to text, ranging from 5% to 8% and images relevant to text ranging from 3% to 7% compared to the approach based on single-embedding schema.

Keywords: deep learning; image-text matching; illustrative images; semantic multi-modal matching; image-text similarity; natural language processing; voting schema

Citation: Golec, J.; Hachaj, T.; Sokal, G. TIPS: A Framework for Text Summarising with Illustrative Pictures. *Entropy* **2021**, *23*, 1614. https://doi.org/10.3390/e23121614

Academic Editor: Mohamed Medhat Gaber

Received: 28 September 2021
Accepted: 26 November 2021
Published: 30 November 2021

1. Introduction

The development of deep neural networks (DNN) has revolutionised issues related to the analysis of images and natural language processing [1]. The ability to generate feature vectors (embedding) from both images and texts has greatly facilitated the semantic analysis of these media. Especially interesting are issues of image-text matching to determine the semantic similarity between them. Image-text matching is an important multi-modal task with a wide range of applications [2]. Research in this area using deep neural networks is relatively new, and many of the relevant results have been published in work from within the last three years.

1.1. State-of-the-Art on Image-Text Matching

Modern methods of comparing and matching text to images are based almost exclusively on deep neural networks [3,4]. We can distinguish several basic issues that determine how to select machine learning methods to make this process efficient and effective. The first is the appropriate choice of architecture and scale of the neural network. Convolutional Neural Networks (CNN) [5] are commonly developed on a fixed resource budget, and then scaled up for better accuracy if more resources are available. Tan et al. [6] systematically studied model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance. They propose a new scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient. Due to the fact that pre-training is a dominant paradigm

in computer vision, [7] image features are commonly retrieved using well-established CNN architectures. The next issue is the application of the image search algorithm. Image Search is a fundamental task playing a significant role in the success of a wide variety of frameworks and applications [8]. An important method to compare semantic similarity between text and images is CLIP Contrastive Language-Image Pre-Training). Conde et al. [9] proposed that a CLIP approach be used for training a neural network on a variety of art images and text pairs, being able to learn directly from raw descriptions about images, or if available, curated labels. Visual attention not only improves the performance of image captioners, but also serves as a visual interpretation to qualitatively measure the caption rationality and model transparency [10]. The use of the CLIP method enabled us to search for illustrative images in which the images do not have ready-made annotations (compare with Joshi et al. [11]). Huang et al. [12] developed an image-text matching approach using a bi-directional spatial-semantic attention network which leverages both the word to regions relation and visual object to words relation in a holistic deep framework for more effective matching. Image-text matching for short (one-sentence) texts has been realised in practice in the Sentence Bert (SBERT) [13] algorithm. It uses the language module BERT [14]. BERT is a language representation model while SBERT generates feature vectors from both image and text and compares them using CLIP. SBERT was developed to optimise previously used solutions such as InferSent [15], Universal Sentence Encoder [16] and SentEval [17]. Vision models trained on multi-modal data sets can benefit from the wide availability of large image-caption data sets. According to [18] CLIP proved to be a reliable method in multi-modal semantic task solving.

Additionally, the latest research findings in the topics of this article can be found in survey papers. Zhu et al. [19] and Rudinac et al. [20] discuss the latest achievements in story summarising. Baltrušaitis et al. [21] and Guo et al. [22] survey the recent advances in multimodal machine learning and present them in a common taxonomy. In reviews [23,24], Gao et al. and Ramachandram et al. present a survey on deep learning for multimodal data fusion to provide readers, regardless of their original community, with the fundamentals of a multimodal deep learning fusion method and to motivate new multimodal data fusion techniques of deep learning. With the rapid growth of social media, users post large volumes of data in various modalities such as text, image, audio, and video. In surveys [25,26], Huddar et al. and Soleymani et al. define sentiment, sentiment analysis, states problems and challenges in multimodal sentiment analysis and finally review some of the recent computational approaches used multimodal sentiment analysis. The Multimodal data-driven approach has emerged as an important driving force for smart healthcare systems. Cai et al. [27] provide a comprehensive survey of existing techniques which include not only state-of-the-art methods but also the most recent trends in the field.

1.2. Study Motivation

In this paper, we propose an algorithm to generate graphical summarising of longer text passages using a set of illustrative pictures, which we refer to as Text Summarising with Illustrative Pictures (TIPS). TIPS is an algorithm using a voting process that uses the results of individual "weak" algorithms to produce a single result. The proposed method includes a summarising algorithm that generates a digest of the input document. Then, a sentence transformer performs text embedding and a group of CLIP similarity-based algorithms trained at image embedding finds semantic distances between the images in the illustration image database and the input text snippets. This is followed by a voting process that extracts the most similar images to the text. Both the TIPS algorithm scheme, the voting process algorithm, and the methodology for evaluating and comparing our algorithm with other image-text matching methods are original achievements presented in this work. Our proposed method matches illustrative images to each sentence that was generated by the summarizer algorithm independently. Therefore, the ability of our algorithm to summarize text using a set of illustrative images relies on the power of the summary-generating algorithm only.

2. Material and Methods

2.1. Text and Image Processing Framework

2.1.1. Text Summarising

Single-document summarising is the task of automatic generation of a shorter document version while retaining its most important information [28]. Currently, neural network-based algorithms trained on relevant language corpora are used to generate text summaries [29–31]. Text summarising typically use a variety of language representation models. Among the most popular and effective models of this type is the BERT (Bidirectional Encoder Representations from Transformers) algorithm [14]. BERT is also based on a neural network, which, among other things, is designed to model the strong connections between words of a language in order to find their representation, which will then be used to generate a summary. Unlike other solutions of this type, BERT is designed to pretrain deep bidirectional representations from unlabelled text.

Text summarizers currently use summarising-specific neural architectures to enhance document-level features. We decided to use the architecture proposed in [28] called BERTSUM. Suppose we have a text $X = [x_1, x_2, \ldots, x_n]$, where $x_i, i \in [1, n]$ are sentences of this text. The task of the summarizer is to select n out of the m sentences that this text consists of ($n \leqslant m$). In this algorithm, a feature vector is generated for each sentence that is part of the text. Then the whole text is processed by Inter-sentence Transformer:

$$\begin{cases} p_0 = B \\ p_i = LN(p_{i-1} + AO(p_{i-1})) \\ p'_i = LN(p(i) + FFN(p(i)) \end{cases} \quad (1)$$

where: B is an output vector of BERT, LN is layer normalisation [32], AO is the attention operation [33] and FFN is feedforward neural network with depth i.

The final layer is a sigmoid classifier. The output from BERTSUM for each of the sentences included in the text generates a value between [0, 1]. The larger the value of the output signal from BERTSUM, the more confident the sentence should be in the summary. The obtained summary $S_x = [x_{j_1}, ..., x_{j_m}]$ is a subset of the original text X and there are m indices $j \in \{1, ..., n\}$.

2.1.2. Sentence Transformer

Sentence transformers are a group of methods for generating a vector of features that describe a sentence (sentence embedding) [34]. A feature of such descriptive vectors is that the algorithms that generate them minimise the distance between vectors of sentences that have similar semantic meaning. Sentence transformer algorithms use a language model to generate a feature vector from a given sentence, which is then the input argument to further processing. In practice, the sentence transformer uses a deep neural network that recalculates the input vectors so that a distance metric can be used to calculate the semantic distance between sentences.

The method proposed in [13] uses the BERT language model, which we described in Section 2.1.1. The output vectors from BERT are then processed with a deep architecture that uses the triplet loss function [35] for training. With triplet loss, the distance between embedding sentences that have similar semantic meaning is minimised during training and the distance between embedding sentences that have different semantic meaning is maximised.

2.1.3. Image Embedding

Most common image embedding solutions are trained as models to classify large data sets of diverse images, for example ImageNet [36]. Networks of this type can also be used to generate feature vectors of images. In this case, the initial convolutional layers of the network are used, which generate a single, typically several hundred dimensional vector describing the input image. Classifier layers of the network are then not used.

The most popular pre-trained implementations of convolutional deep neural networks are VGG16 [37], ResNet50 and its modifications [38], InceptionV3 [39] or MobileNet [40].

2.1.4. Image-Text Matching

Image-text matching is a group of methods that allows evaluating the semantic similarity between text and image content by measuring this similarity using a given metric. Modern algorithms of this type use sets of methods that we discussed in earlier sections and have much in common with image classification problems. Image classification, which boils down to assigning an image to one of a predefined class, is a very well-studied and widely applied problem, as we pointed out in Section 2.1.3. The problem of determining image class can also be considered more broadly by using methods that will automatically generate descriptions of the contents of images. This is currently done using deep models that combine image embedding with the ability to generate descriptions using recurrent Long short-term memory (LSTM) neural layers such as [41–43]. A large survey on this field can be found at [44,45]. Thus, one could use such automatically generated image descriptions with sentences that have been generated by the text summarizer (see Section 2.1.1) to determine the similarity between the text and the various images we have, in order to select the image that minimises the given distance metric [46]. However, this approach has a major drawback: the quality of image descriptions generated by automatic algorithms is not yet perfect, and semantic comparison of texts is a complex issue. This results in a build-up of errors generated by both approaches, which can significantly affect the quality of the entire text-to-image matching method. For this reason, dedicated and specially trained solutions are used for image-text matching, whose individual components originate from the areas of natural language processing, text embedding and image embedding.

A Contrastive Language-Image Pre-training (CLIP) method based on neural network architecture has been is proposed in the paper [47] to evaluate the similarity between text and image. In order to perform CLIP, a similarity assessment algorithm learns a multi-modal embedding space by jointly training an image encoder and text encoder to maximise the cosine similarity of the image and text embedding. The utilised batch construction technique used in CLIP is the multi-class N-pair loss [48]. Image-text similarity is calculated as follows:

$$\begin{aligned} I_e &= L2(I_f \circ W_I) \\ T_e &= L2(T_f \circ W_T) \\ sim_{clip} &= (I_e \circ T_e) \cdot e^p \end{aligned} \tag{2}$$

where: $L2$ is euclidean norm, $\circ$ is a dot product, I_f is image features (embedding), T_f is text features (embedding), W_I is learned projection of image to embedding (to be trained during CLIP model fitting), W_T is learned projection of text to embedding (to be trained during CLIP model fitting), p is a learning rate. The training procedure of CLIP is based on minimisation of cross entropy loss. CLIP utilises Transformer text encoder (embedder) [49] and image embedding algorithm (i.e., Vision Transformer image encoder (embedder) [50], ResNet etc.).

2.2. Data Sets

In this subsection we describe the data set that we have utilised in our research.

2.2.1. Text Data Set

The text dataset was obtained from the Brunel University London website (https://brunel.figshare.com/articles/dataset/4000_stories_with_sentiment_analysis_dataset/7712540, (accessed on 27 September 2021)). The study used 426 short stories that are equal to or less than 1000 characters in length. Due to this we will call this data set Stories426. The collection consists of humorous stories or short tales with a moral. The authors include Aesop, Ambrose Bierce, James Baldwin or Kate Chopin. All texts have been summarised using a BERTSUM (see Section 2.1.1 for exact information), then divided into sentences. This data set was chosen because of its use of natural language and the variety of stories.

An additional advantage, is that the sentences are written correctly in terms of style and language. Obtaining linguistically correct texts is more difficult with data from social media platforms because their users often use verbal abbreviations, e.g., 'LOL' which means laugh out loud, 'U' which is equivalent to 'you', do not use complete 'sentences or forget about linguistic correctness. For the stories used, the full story is described, which helps in creating a full illustration for the text. A person can also easily verify if the proposed images match the content they are paired with. Another advantage of this data set is that it is freely available to the public.

2.2.2. Image Data Set

The image data set was built from nearly 25,000 (24,996) publicly available nature-themed images, sourced from the Unsplash platform (https://unsplash.com/ (accessed on 27 September 2021)). Embedding for this set of images was generated using several deep neural networks. We have used a 50-layer residual network (ResNet 50, RN50), 101-layer residual network (ResNet 101, RN101), and a four times scaled RN50 according to the EfficientNet scaling rule [6]. We also used the Vision Transformer image encoder mentioned in Section 2.1.3 in order to generate embedding for two of its architectures: ViT-B32 (ViT32) and ViT-B16 (ViT16). The Vision Transformer image encoder, in comparson to state-of-the-art DNN models we mentioned in Section 2.1.3, requires substantially fewer computational resources to train. We have used the pre-trained web weights provided at https://github.com/openai/CLIP/blob/main/clip/clip.py (accessed on 27 September 2021). The use of Transformer text embedding [49] and ViT32 image embedding is the same as the SBERT solution architecture [13].

2.3. Proposed Method for Text Summarising with Illustrative Pictures (TIPS)

In Figure 1, we present the data processing pipeline of our proposed algorithm for generating a graphical text summary using illustrative images. It is an algorithm using a properly designed voting process that uses the results of individual "weak" algorithms to produce a single result.

The first step is to prepare a text summarizer using a summarising algorithm. For this purpose, we use the BERTSUM algorithm described in Section 2.1.1. Then, using the sentence transformer, we perform text embedding using the Transformer algorithm [47]. As a set of illustration images $O = \{o_1, o_2, ..., o_r\}$ (r is image count of the data set that contains all potential illustrative images) should contain diversified set of images with a wide range of topics if the texts to be analysed are to cover a wide range of topics. If we assume that texts are to cover a narrower range of specialised topics we use a set of images that are characteristic to those very topics i.e., architecture, sport events etc.

Let us first consider the performance of a single image-text matching algorithm. The algorithm uses the given image embedding method (a suitable deep neural network, see Section 2.1.3). The input text is processed by the sentence transformer algorithm (see Section 2.1.2). The image and text feature vectors are used to train CLIP (see Section 2.1.4). This training only needs to be done once. Assuming that we have a given image database to serve as a source of illustrative images, we can perform an equal embedding of this entire database using image embedding, which is part of the given image-text matching method. For the given database, this process also takes place once.

After the text summary is performed, each sentence extracted from the text is processed by a set of image-text matching algorithms (see Section 2.1.4). Image-text matching is performed for each sentence in the text summary separately. For each summary, a vector is generated which coordinates are numbers that are proportional to the semantic similarity between the text and the illustration: see Equation (2). Each sentence will be represented by an illustration image for which the value of (2) reaches a maximum.

$$\forall x_k \in S_x \rightarrow o_l : max_l(sim_{clip}(o_l, x_k)) \quad (3)$$

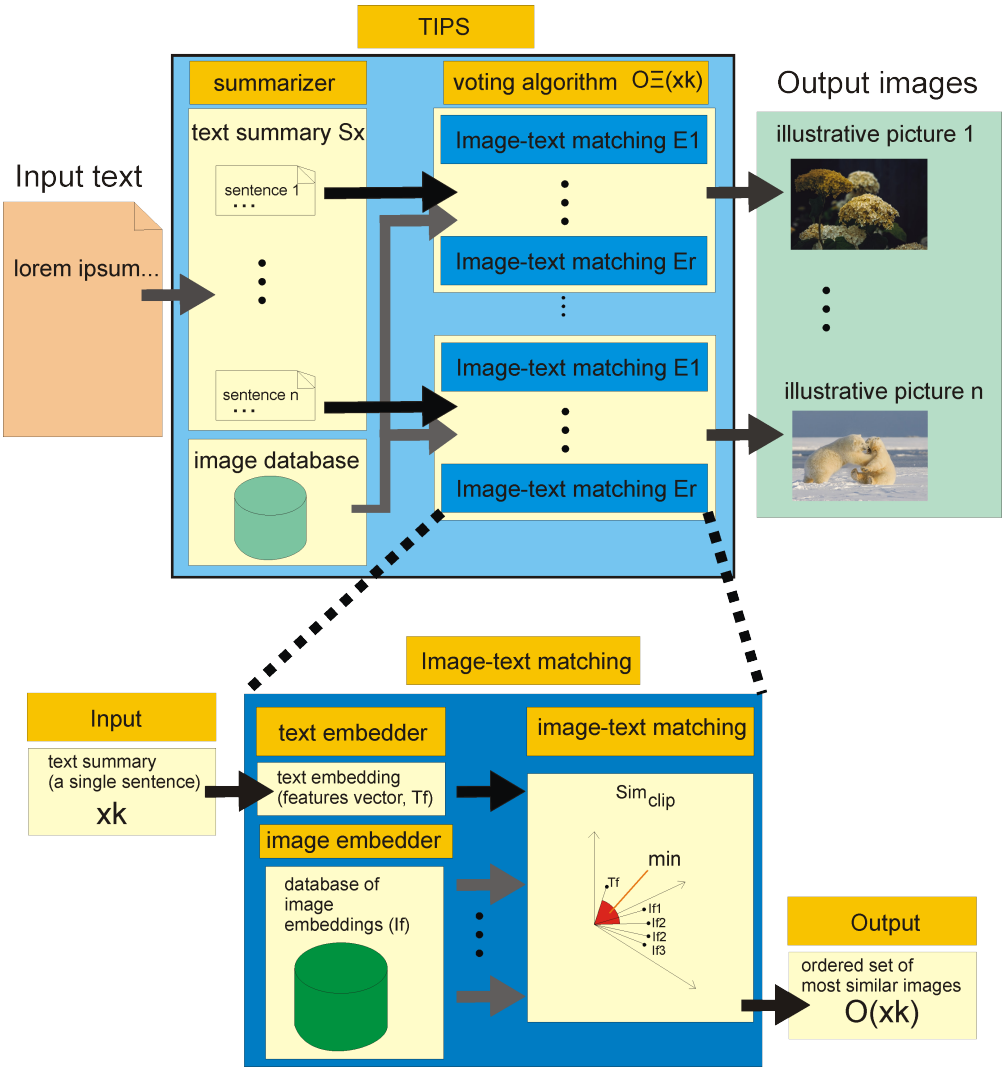


Figure 1. This figure presents the pipeline of the proposed algorithm for text summarising with illustrative pictures (TIPS) algorithm.

Evaluation of the quality of image to text assignment is performed using similarity sim_{clip}.

Let:

$$O(x_k) = ord(O, sim_{clip}(o_l, x_k)) \tag{4}$$

be a set of images O ordered by sim_{clip} between sentence x_k and image $o_l \in O$ in descending order. In other words $O(x_k)$ is a set of images where first element has the highest value of sim_{clip} with x_k and the last element has the smallest value of sim_{clip} with x_k.

The ordering (4) depends on the image embedding E we have used. For this reason, we can write more generally:

$$O^E(x_k) = ord(O, sim^E_{clip}(o_l, x_k)) \tag{5}$$

Let $O^E(x_k)_t$ be an ordered subset of the set (5), consisting of t initial elements of $O^E(x_k)$. With r different embedding methods defined $Xi = E_1, ..., E_r$ we can propose an algorithm that will use (5) to create a voting scheme. The goal of this scheme will be to order the sets of images proposed by each embedding in Ξ order according to the sum of the normalised sim^E_{clip}:

$$O^{\Xi}(x_k)_t = \bigcup_{\Xi} O'^E(x_k)_t \tag{6}$$

In the case of $O'^{E}(x_k)_t$, the similarity between x_k and o_l is computed as:

$$sim'^{E}_{clip}(x_k, o_l) = \frac{sim^{E}_{clip}(x_k, o_l)}{\sum_{s=1}^{t} sim^{E}_{clip}(O(x_k)_s, o_l)} \tag{7}$$

That means that the similarities between x_k and o_l for a given $E \in \Xi$ are divided by the sum of the similarities of the first p similarities between $O(x_k)_s$ and o_l ordered according to (5). This operation is done to pseudo-normalise sim^{E}_{clip} so that the individual orderings of O'^{E}_{t} have comparable ranges of sim^{E}_{clip} values. Thus, the ordered set $O^{E}(x_k)_t$ consists of the sum of the elements included in the individual ones of $O'^{E}(x_k)_t$. The criterion for ordering the images o_l that are part of $O'^{E}(x_k)_t$ is the sum of the (7) that has been assigned to a given o_l by the individual $E \in \Xi$. If any $O^{E}(x_k)_t$ does not contain o_l, then we assume that for this E $sim'^{E}_{clip}(x_k, o_l) = 0$.

In the simplest case where $t = 1$ voting schema (6) works like majority voting schema. In case two or more images received the same number of votes as the most similar to the given text, one of these most similar items is returned randomly.

As the value of t increases, the voting algorithm (6) will propose $O^{\Xi}(x_k)_t$, which will contain an increasing number of proposed images. The order of these images might differs. Of course, it is always true that:

$$O^{\Xi}(x_k)_p \subset O^{\Xi}(x_k)_{t+1} \tag{8}$$

This means that all images that are in the set $O^{\Xi}(x_k)_t$ for a given p are also in the set $O^{\Xi}(x_k)_{t+1}$.

The above Equation (7) defines our proposed text summarising with illustrative pictures (TIPS) method. In summary, for the text X:

$$X \xrightarrow{TIPS} \{o_{k_1}, \ldots, o_{k_n}\} \tag{9}$$

We can also write (9) as:

$$TIPS(X) = \{o_{k_1}, \ldots, o_{k_n}\} \tag{10}$$

Summarizer turns paragraph text into sentences and then passes the tokenized sentences to the BERT model for inference to output embedding. That embedding is then clustered with K-Means in order to select sentences that are closest to the centroid. Those closest sentences are candidates for summary [51]. Due to this fact the summary consists of sentences that are already present in the original text.

The principles of the CLIP method in practice limit its effectiveness to single sentences, so it would be ineffective to use it for the entire text [47]. It is possible that embedding of the certain sentences will not create "spherical" clusters with representative centroid. In that scenario the "centre" sentence might be not representative for the whole text, however already published papers proved for some extend that application of K-means clustering for finding most important sentences resulted in relatively high ROUGE scores in comparison to other approaches [52–54]. Due to this fact we did not evaluate the quality of the obtained summary.

2.4. Evaluation and Comparison of Image-Text Matching Algorithms

Evaluation of image-text matching in terms of semantic similarity of large texts is a very difficult issue because we do not have a data set for which there would be a ground truth. Additionally, if we want to compare results obtained by two or more different image-text matching algorithms that use the clip similarity measure, we need to remember that each of these algorithms is trained independently, and it would be incorrect

to compare these similarity measures directly. For this reason, we have proposed a number of coefficients that are useful for evaluating and comparing image-text matching algorithms.

While we cannot directly compare sim_{clip} values between methods, we can examine the semantic distances returned between successive recommended images:

$$\begin{cases} m_{1,2}(x_k) = sim_{clip}(O(x_k)_1, x_k) - sim_{clip}(O(x_k)_2, x_k) \\ m_{1,3}(x_k) = sim_{clip}(O(x_k)_1, x_k) - sim_{clip}(O(x_k)_3, x_k) \end{cases} \tag{11}$$

where $O(x_k)_1$ is a first element in $O(x_k)$, $O(x_k)_2$ is a second element in $O(x_k)$ etc. Due to this $m_{1,2}(x_k)$ is a difference between sim_{clip} value of the most similar image and the second most similar image. A high value of this index may indicate that the image data set is diverse as well as that the selected most similar image is significantly more similar to the text than the other images that are in the image set while using certain E. The average value of these indexes can also be counted for the entire test set TS:

$$\begin{cases} \overline{M_{1,2}} = \frac{\sum_{X \in TS} \sum_{x_k \in S_X} m_{1,2}(x_k)}{\#(\sum_{X \in TS} n_X)} \\ \overline{M_{1,3}} = \frac{\sum_{X \in TS} \sum_{x_k \in S_X} m_{1,3}(x_k)}{\#(\sum_{X \in TS} n_X)} \end{cases} \tag{12}$$

where $\#(\sum_{X \in TS} n_X)$ is the cardinal number of the set, that are summaries of each text X contained in the set TS. Equation (11) are statistics for a single text. Equation (12) are statistics for the entire set ST.

We can also examine the common part of the set of recommended illustration images for all algorithms included in Ξ:

$$In(\Xi, t) = \frac{\#(\forall x_k \in S_x : (\bigcap_{\Xi} O'^E(x_k)_t)}{n} \tag{13}$$

as well as the sum of such sets:

$$Un(\Xi, t) = \frac{\#(\forall x_k \in S_x : (\bigcup_{\Xi} O'^E(x_k)_t)}{n} \tag{14}$$

Obtained values inform us how much variation there is in the t first recommendations of each of the algorithms included in Ξ.

Another statistic is the cardinality of the set composed of common part of set of images recommended by (6) for t and set of images recommended by (6) for $t+1$ divided by the number of all summaries $TIPS(X)$.

$$V(\Xi, t) = \frac{\#(\forall x_k \in S_x : sim_{clip}(O^{\Xi}(x_k)_t 1, x_k) = sim_{clip}(O^{\Xi}(x_k)_{t+1} 1, x_k))}{n} \tag{15}$$

The above equation determines how much successive votes are consistent with each other, and can be thought of as a way of determining the stability of the voting process as well as the consistency of the recommendations returned by each component $O'^E(x_k)_t$. It can be seen that $V(\Xi, t) \in [0, 1]$. The value of $V(\Xi, t)$ equals 0 when:

$$\forall x_k \in x_k : sim_{clip}(O^{\Xi}(x_k)_t 1, x_k) \neq sim_{clip}(O^{\Xi}(x_k)_{t+1} 1, x_k) \tag{16}$$

The value of $V(\Xi, t)$ equals 1 when:

$$\forall x_k \in x_k : sim_{clip}(O^{\Xi}(x_k)_t 1, x_k) = sim_{clip}(O^{\Xi}(x_k)_{t+1} 1, x_k) \tag{17}$$

Counterintuitively, the phenomenon where $V(\Xi, t) = 1$ is not necessarily an advantageous situation. This situation means that every $E \in \Xi$ always returns an identical first image recommendation, which may imply little diversification in the embedding methods used.

A set of following statistics is also worth investigating because they report the effect of the t parameter on the performance of the TIPS algorithm (this will be discussed in Section 4):

$$\begin{cases} sim_{min}(X) = min(TIPS(X)) \\ sim_{max}(X) = max(TIPS(X)) \\ sim_{mean}(X) = mean(TIPS(X)) \\ sim_{med}(X) = med(TIPS(X)) \end{cases} \quad (18)$$

where: min is minimal, max is maximal, mean is mean and med is median value of sim_{clip} among all image-text matching calculated by TIPS.

We have also added an evaluation of the obtained image recommendation results by three human judges. One of the evaluators was a co-author of this paper, (T.H.) and two others were persons not directly related to the research presented in this paper and without a background in computer science. To each judge, the computer program presented a sentence generated by the summary algorithm and six illustrative images selected by TIPS and using non-voting single-embedding schema (4) in which E was ViT16, ViT32, RN50, RN101, and RN50x4. Those six illustrative images were displayed in a random order so as to eliminate the situation where the judge assigned some meaning to the order of the images. Judges were informed of the purpose of the study and that the images would be presented in random order. Each judge independently assessed the relationship between the text and each of the six illustrative photographs using a three-point judging rating scale (JS) in the range [0–2] according to the subjective impressions. Each judge made evaluation separately without contacting two others. This approach is similar to [55]:

- JS = 0: the image is not relevant to the text;
- JS = 1: the image is relevant to the text;
- JS = 2: the image is highly relevant to the text.

In the next section, we will present the evaluation results of our method on the data sets discussed in Section 2.2.1.

3. Results

In order to find illustrative images for each S_X according to the TIPS method described in Section 2.3, we have prepared an implementation of the proposed solution.

Our proposed method for finding illustrative images for text was implemented in Python 3.5. In order to generate S_X for each of the short stories included in the Stories426 collection described in Section 2.2.2, we implemented the method described in Section 2.1.1 based on the solution proposed by Miller [51]. For this purpose we used the libraries spaCy 3.1, Transformers 4.1, NeuralCoref 4.0, Summarizer 0.0.7, Sentencepiece 0.1.96, pytorch_pretrained_bert 0.6.2. Computations were performed on the Google Colab platform using Torch 1.6 computational libraries.

Embedding of text was done using the Transformer algorithm [49] using the implementation of https://sbert.net/docs/package_reference/SentenceTransformer.html (accessed on 27 September 2021). Embedding of images was done using the methods described in Section 2.1.3 and CLIP similarity computation was done using the algorithm described in Section 2.1.4. We used Pytorch 1.7 and Torchvision cudatoolkit 11. The source codes and data sets of the programs we prepared can be downloaded from https://github.com/JusMia/TIPS (accessed on 27 September 2021).

We have made recommendations using the illustrative images presented in Section 2.2.2 for each text that was in the Stories426 using the TIPS algorithm utilising the voting schema (6). We used the image embedding methods ViT16, ViT32, RN50, RN101, and RN50x4 in the voting algorithm $O^{\Xi}(x_k)_t$ (6). We performed calculations for p values with an interval of $[1, 100]$. For the purposes of evaluation we have also calculated illustrative image recommendations using non-voting single-embedding schema (4), in which E was ViT16, ViT32, RN50, RN101, and RN50x4.

Figure 2 shows the values of $V(\Xi, t)$ for different p. The separate results were connected by the segments marked in black. We smoothed the resulting plot using spline

approximation. We have marked the spline in green. In red we marked four local maxima of the smoothed function $V(\Xi, t)$. For the corresponding values of p we will perform a more detailed analysis later on.

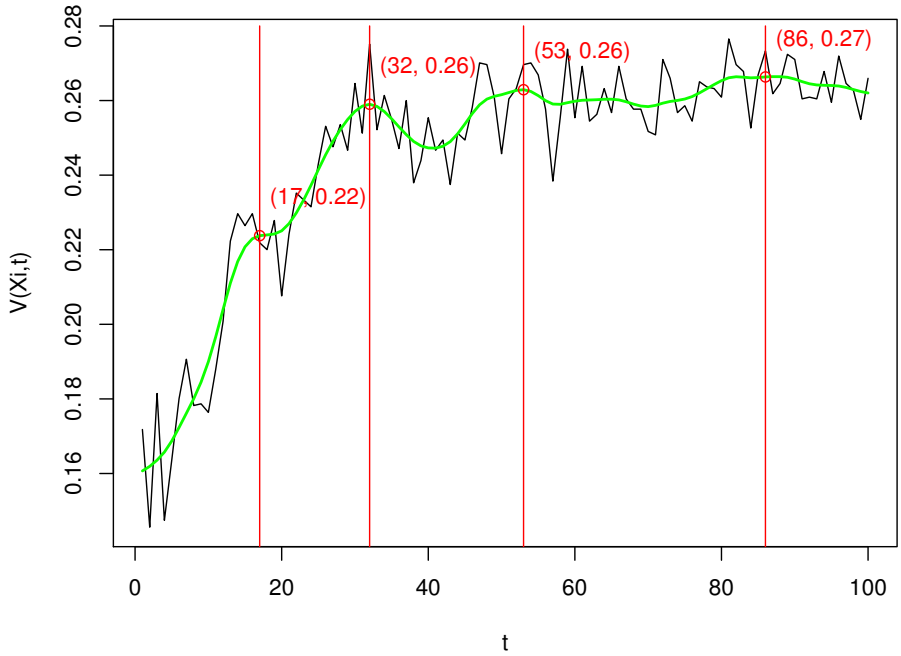


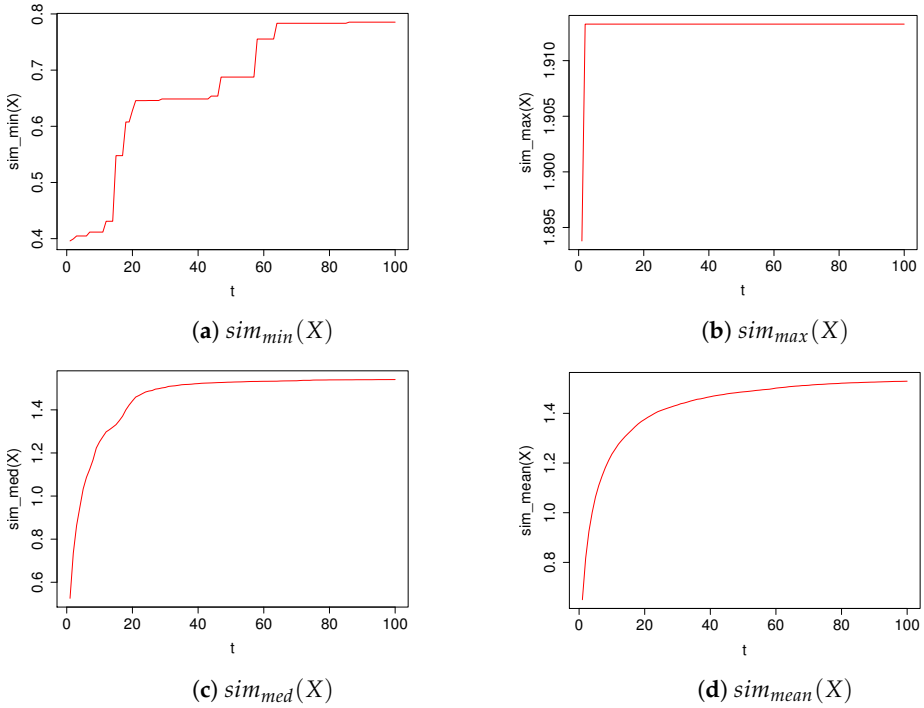
Figure 2. Plot of $V(\Xi, t)$ values for different p of the TIPS algorithm. The individual results have been connected by the segments highlighted in black. We smoothed the resulting plot using spline approximation. We have marked the spline in green. In red we have marked four local maxima of the smoothed function $V(\Xi, t)$.

In Table 1, we presented values of coefficients $\overline{M_{1,2}}$ and $\overline{M_{1,3}}$ (12) for recommendations $O^E(x_k)$ where embedding E used algorithms ViT16, ViT32 (equivalent to the SBERT method), RN50, RN101, RN50x4, and our proposed TIPS algorithm with t-values equal to 17, 32, 53, and 86, which were the local maxima detected previously (see Figure 2).

Table 1. Coefficient values of $\overline{M_{1,2}}$ and $\overline{M_{1,3}}$ (12) for the recommendation $O^E(x_k)$ where embedding E used algorithms ViT16, ViT32 (equivalent to the SBERT method), RN50, RN101, RN50x4, and our proposed TIPS algorithm with p-values of 17, 32, 53, and 86.

Method	$\overline{M_{1,2}}$	$\overline{M_{1,3}}$
ViT16	0.017	0.026
ViT32 (SBERT)	0.017	0.026
RN50	0.021	0.032
RN101	0.01	0.015
RN50x4	0.013	0.02
TIPS $p = 17$	0.013	0.03
TIPS $p = 32$	0.008	0.015
TIPS $p = 53$	0.005	0.011
TIPS $p = 86$	0.004	0.008

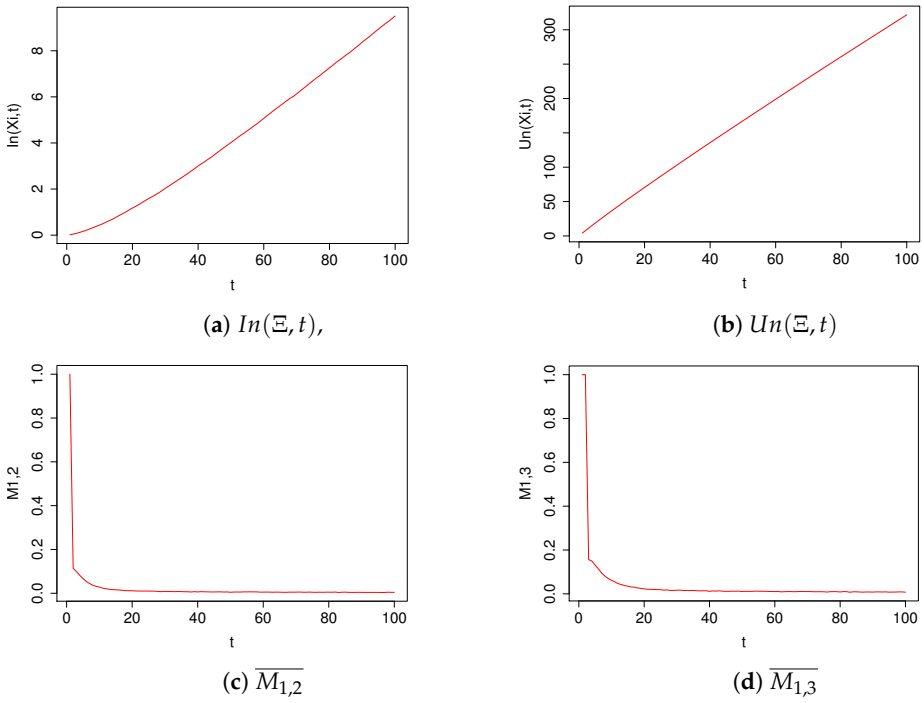
In plots in Figure 3 we have shown the values of the coefficients $sim_{min}(X)$, $sim_{max}(X)$, $sim_{mean}(X)$, $sim_{med}(X)$ for the TIPS algorithm with parameter values t in the range $[1, 100]$. In plots in Figure 4 we have shown the values of the coefficients $In(\Xi, t)$, $Un(\Xi, t)$, $\overline{M_{1,2}}$ and $\overline{M_{1,3}}$ for the TIPS algorithm with parameter values t in the range $[1, 100]$.



(**a**) $sim_{min}(X)$ (**b**) $sim_{max}(X)$

(**c**) $sim_{med}(X)$ (**d**) $sim_{mean}(X)$

Figure 3. Performance of the TIPS algorithm for different values of the parameter t.



(**a**) $In(\Xi, t)$, (**b**) $Un(\Xi, t)$

(**c**) $\overline{M_{1,2}}$ (**d**) $\overline{M_{1,3}}$

Figure 4. Performance of the TIPS algorithm for different values of the parameter t.

In Table 2, we have presented the exact values of the coefficients of the TIPS algorithm shown in Figures 3 and 4 for the selected t values.

Table 2. The exact values of the coefficients of the TIPS algorithm shown in Figures 3 and 4 for the selected t values.

t	$In(\Xi, t)$	$Un(\Xi, t)$	$sim_{min}(X)$	$sim_{max}(X)$	$sim_{med}(X)$	$sim_{mean}(X)$
1	0.02	4.13	0.40	1.89	0.53	0.65
2	0.05	7.93	0.40	1.91	0.74	0.82
3	0.08	11.69	0.40	1.91	0.86	0.93
17	0.93	60.53	0.55	1.91	1.37	1.35
22	1.33	77.22	0.65	1.91	1.47	1.39
52	4.23	173.77	0.69	1.91	1.53	1.49
86	7.91	279.14	0.79	1.91	1.54	1.52

In Figure 5 we have shown example of the illustrative image recommendation results for three example text summaries x_k. Images are calculated as $O^E(x_k)_1$ for the embedding algorithms ViT16, ViT32, RN50, RN50x4, RN101 and $O^{\Xi}(x_k)_1$ for TIPS, $p = 32$.

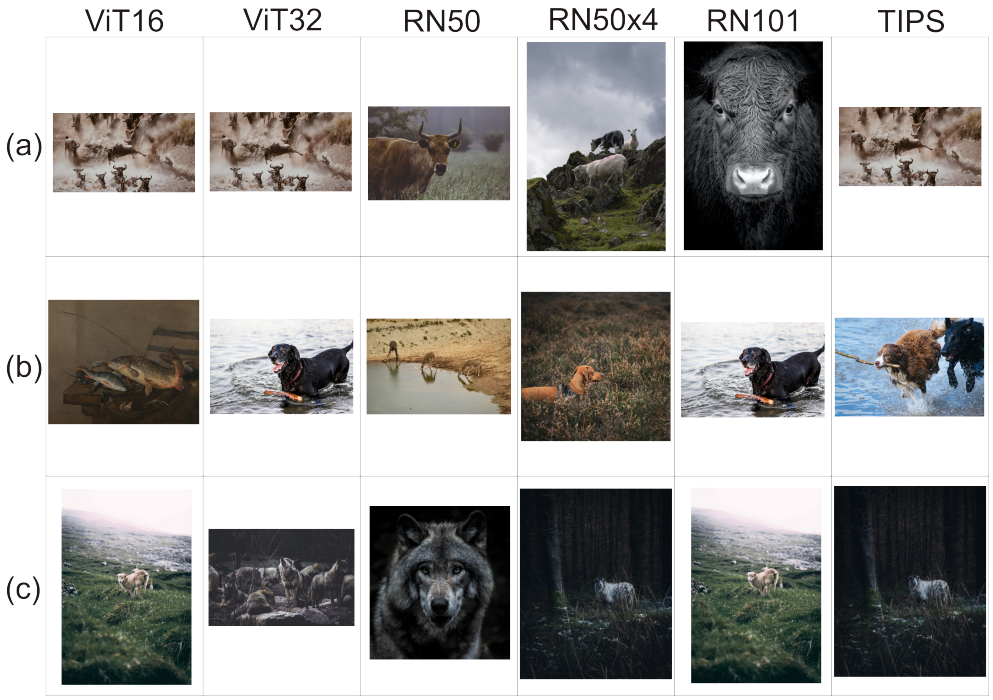


Figure 5. Sample illustrative image recommendation results for three sample text summaries x_k. The texts are: (**a**) "A Bull once escaped from a Lion by entering a cave which the Goatherds used to house their flocks in stormy weather and at night"; (**b**) "A fine hide makes an excellent meal for a hungry Dog, but the water was deep and the Dogs could not reach the hides from the bank"; (**c**) "The next day, dressed in the skin, the Wolf strolled into the pasture with the Sheep".

For the evaluation based on human judges we used half of all the texts from the Stories426 collection (exactly 1089 one -sentence summaries). We did not conduct the evaluation on the entire dataset because the work of each human judge took several hours and half of the dataset should be sufficient to obtain statistically representative results. In Table 3 and Figure 6, we presented the JS averaged over each illustrative image recommendation algorithm. In Table 4, we also counted the correlation between the JS scores obtained by each recommendation algorithm. In this way, we investigated whether there is a correlation between the scores of each algorithm and the semantic interpretation of their relevance by human judges. To count this correlation matrix, we used the JS scores of all judges simultaneously. In Table 5, we presented the results of the correlation analysis between the JS of the individual judges. We counted this correlation matrix to investigate whether there is a correlation between the scores of individual judges. In Figure 7, we

presented the distribution of JS ratings given by each judge to each descriptive image recommendation algorithm.

Table 3. The JS averaged over each illustrative image recommendation algorithm.

	Judge 1	**Judge 2**	**Judge 3**
TIPS	0.96	0.98	0.82
ViTB32	0.74	0.81	0.66
ViTB16	0.74	0.80	0.67
RN101	0.68	0.75	0.62
RN50x4	0.71	0.77	0.61
RN50	0.69	0.75	0.58

Table 4. Correlation between the JS scores obtained by each recommendation algorithm.

	TIPS	**ViTB32**	**ViTB16**	**RN101**	**RN50x4**	**RN50**
TIPS	1.00	0.55	0.52	0.55	0.57	0.52
ViTB32	0.55	1.00	0.47	0.46	0.47	0.45
ViTB16	0.52	0.47	1.00	0.44	0.45	0.41
RN101	0.55	0.46	0.44	1.00	0.44	0.44
RN50x4	0.57	0.47	0.45	0.44	1.00	0.48
RN50	0.52	0.45	0.41	0.44	0.48	1.00

Table 5. Correlation analysis between the JS of the individual judges.

	Judge 1	**Judge 2**	**Judge 3**
judge 1	1.00	0.76	0.77
judge 2	0.76	1.00	0.66
judge 3	0.77	0.66	1.00

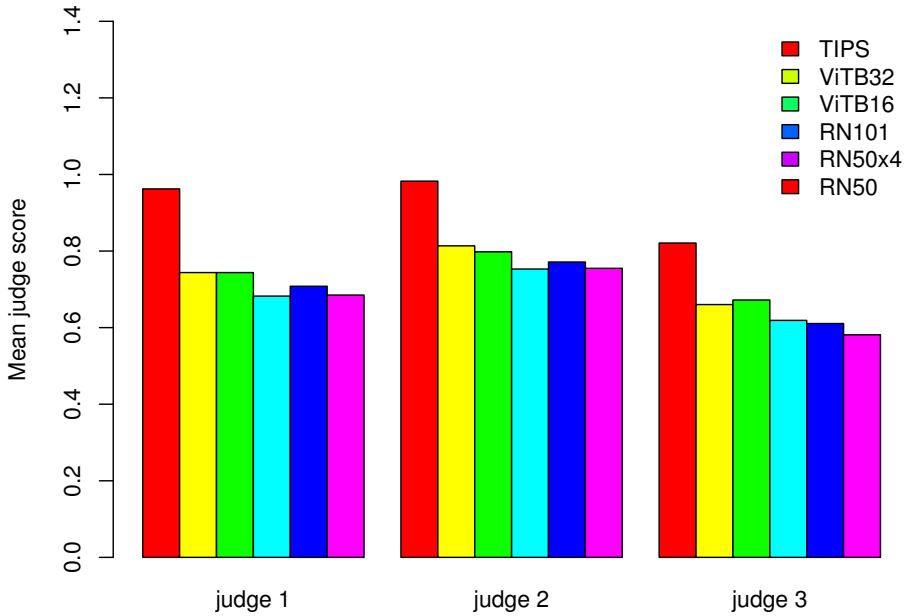
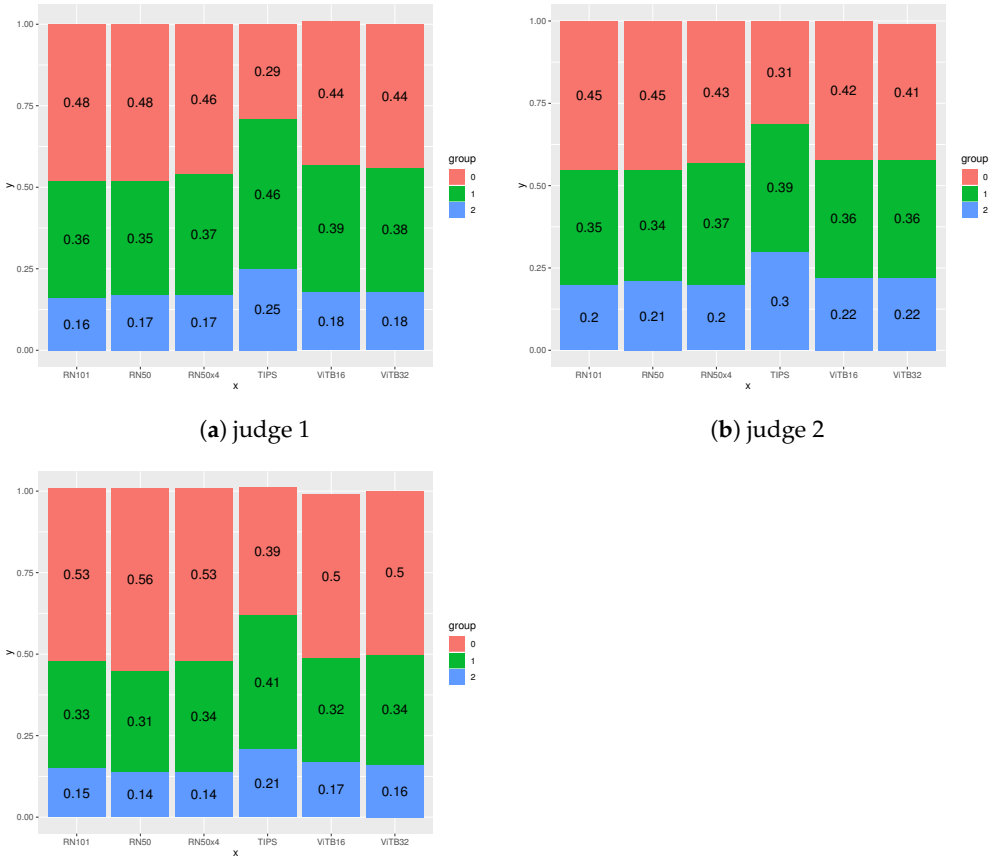


Figure 6. JS averaged over each illustrative image recommendation algorithm (see Table 3).



(**a**) judge 1

(**b**) judge 2

(**c**) judge 3

Figure 7. Distribution of JS ratings given by each judge to each descriptive image recommendation algorithm.

4. Discussion

Unfortunately, we have no way to directly compare the results of the similarity function sim_{clip} computed by each of the algorithms we test, because there are no ground truth values. For this reason, we used the values of $sim_{M_{1,2}}$ and $sim_{M_{1,3}}$ to compare the recommendations of each method.

As can be clearly seen in Equation (11) the TIPS approach with t > 17 has smaller values of $\overline{M_{1,2}}$ and $\overline{M_{1,3}}$ than all other considered algorithms. This fact means that, respectively, the first two or three recommendations given by the TIPS algorithm contain images that are more similar to each other according to the CLIP distance than is the case for algorithms that do not use the voting scheme. According to Figure 4c,d, these values for TIPS decrease exponentially as the parameter t increases. This result should be considered in conjunction with the plot of the values of $In(\Xi, t)$ and $Un(\Xi, t)$ in Figure 4a,b. As can be seen, an increase in the parameter p results in a much slower increase in the common portion of the images recommended by all TIPS component algorithms relative to the sum of the image sets recommended by these component algorithms. This means that each of the component algorithms based on a different image embedding algorithm E that we considered in our study proposed a differentiated set of images $O^E(x_k)_t$. The obtained values of $\overline{M_{1,2}}$, $\overline{M_{1,3}}$, $In(\Xi, t)$, and $Un(\Xi, t)$ indicate that, for the image set we used, the TIPS algorithm allows the integration of the best (highest scored) results of the different recommendation algorithms by diminishing the influence of images that are a disjointed part of the recommendations of the component algorithms. The values of $In(\Xi, t)$ and $Un(\Xi, t)$ increase in a monotonic, near linear fashion.

The shape of the variation of the value of $V(\Xi, t)$ as a function of t is close to logarithmic, which confirms the great diversity when it comes to the recommendations made

by the algorithm included in Ξ. We have chosen to highlight a few local maxima that become visible in the spline-smoothed graph and present detailed computations for TIPS at these points in Table 2. Note, however, that the occurrence of these maxima is not universal and may vary depending on the set of illustrative images and text data used. According to the results in Table 2 and Figure 3, the shape of variation of $sim_{min}(X)$ has a step-function character as the parameter t increases. This means that the recommendation of illustrative images that are among the first t highly scored values returned by the selected component recommendation algorithm might also be recommended by the other component algorithms. This illustrates the situation that, given a certain margin t, the component recommendation algorithms return a similar common portion of the best matching images. The value of $sim_{max}(X)$ is also spiking, but it reaches its maximum value much faster than $sim_{min}(X)$. The variability of the coefficients $sim_{med}(X)$ and $sim_{mean}(X)$ have an increasing character close to logarithmic and are very similar in shape. They represent some intermediate state between the extreme statistics $sim_{min}(X)$ and $sim_{max}(X)$. The entire set of graphs shown in Figure 3 shows that as the parameter p increases, the TIPS algorithm recommends illustrative images with increasing similarity values until it comes to maximising the values of the individual statistics based on sim_{clip}. This fact and the shape of plots of $sim_{med}(X)$ and $sim_{mean}(X)$ prove that the t parameter is a predictable scaling factor of the confidence range of the recommendation result obtained by TIPS.

Figure 5 visualises examples of the recommendations proposed by the individual component algorithms and the TIPS algorithm for t = 32. Full texts of those stories can be downloaded from https://github.com/JusMia/TIPS/tree/main/stories (accessed on 27 September 2021).

The value t = 32 was chosen because it was the first local maximum $V(\Xi, t)$ for which the values $\overline{M_{1,2}}$ and $\overline{M_{1,3}}$ of TIPS performed better than the individual component algorithms (see Table 1). We can see that each of the illustrative images proposed for texts by the individual component algorithms has some features that make it similar to the given text. An interesting case is sentence (b), in which ViT16 and RN50 seem to be very loosely related to the text and do not capture the semantic meaning of it. It is noteworthy that the TIPS recommendation $O^{Xi}(x_k)_1$ need not be among the $O^{E}(x_k)_1$ of the individual component algorithms. The TIPS recommendation t = 32 for sentence (a) is the same as the recommendation for E = ViT16 and E = Vit32 and for sentence (c) for E = RN101. In the case of sentence (b), TIPS t = 32 proposed a different illustrative image, which is semantically consistent with the content of the sentence but not in the set of first recommendations of the individual component algorithms.

The evaluation performed by human judges confirmed that the average JS value is the highest for the TIPS algorithm: see Table 3 and Figure 6. The mean JS for TIPS is 0.96, 0.98 and 0.82 while for a non-voting single-embedding schema (4) with E ViTB32 it was 0.75, 0.81 and 0.66. Also, the results shown in Figure 7 confirm that individual judges are more likely to rate the recommendations made by TIPS as being relevant or highly relevant to the text. Human judges found 25%, 30% and 21% TIPS recommendations highly relevant and 46%, 39% and 41% relevant to text. In the case of non-voting single-embedding schema (4) with E ViTB32 18%, 22% and 16%, recommendations were found highly relevant and 38%, 36% and 34% recommendations were found relevant. According to the results in Table 4, the average value of JS is medium positively correlated between all considered algorithms. Correlation varies from 0.41 to 0.57. This is an important found because it indicates that from the judges perspective all considered algorithms work in similar manner proposing correlated images to subject of the text. As can be seen in Table 5, there is a moderate (0.66 between judge 2 and judge 3) and strong positive correlation (0.76 and 0.77) between judges JS. That means that all three human judges have evaluated algorithms results in a similar manner. The evaluation results showed some limitations of the proposed method due to the fact that it works on a fixed set of image data. In our case the Unsplash Lite dataset nature-themed images might be not suitable for some topics. The judges estimated that between 29% and 39% of the images selected by TIPS were not relevant to the text.

However, this is a much better result than when using the non-voting single-embedding schema where the number not relevant to the text was between 41% to even 56%.These results show that using a method based on voting schema allows us to increase the quality of matching evaluated with JS.

5. Conclusions

Designing algorithms to suggest illustrative images for text is a very advanced and relatively new research topic. Based on the results presented in the previous section of our paper, we can conclude that our proposed algorithm allows us to make suggestions of illustrative images that are to some extent semantically consistent with the content of the summaries' sentences of the text to which they relate. The recommendation quality of the TIPS algorithm is a function of the set of illustrative images that have been selected to illustrate the text and the efficiency of sub-recommendation algorithms used by TIPS in the voting schema. An equally important aspect of TIPS is the algorithm that performs text summarization. Since the selection of illustrative images is based on applying a voting schema to the individual sentences included in the summary, this represents a potential weakness in our algorithm if unrepresentative sentences are selected. Regardless, voting schema for individual sentences proved to be a very effective approach. In this paper, we presented a number of coefficients that can be used to test whether a recommendation algorithm $O^{Xi}(x_k)_t$ meets its expectations. We have done this by presenting some benchmark solutions based on a Stories426 text database and a set of images described in Section 2.2.2.

Based on the evaluation of the results conducted by the three human judges, it can be concluded that the use of the TIPS voting scheme increased the accuracy of matching illustrative images to texts. The judges estimated that the use of TIPS resulted in an increase in matching highly relevant images to text ranging from 5% to 8% and relevant to text ranging from 3% to 7% compared to the approach based on single-embedding schema with E ViTB32, which was the best evaluated algorithm from the single-embedding schema group.

We have shown that the TIPS algorithm allows the integration of the best (highest scored) results of the different recommendation algorithms by diminishing the influence of images that are a disjointed part of the recommendations of the component algorithms. Our computations and experiments can be replicated as we publish the full source codes of both the TIPS algorithm and the entire evaluation process of our approach.

There are many more available image datasets, including common sense ones which could be useful for story understanding in further research including COCO [56] and the Open Images Dataset [57].

An issue that we believe is worthy of future investigation is the feasibility of applying text-based image generation methods at the end of the image processing pipeline. We anticipate that the use of a sufficiently large database of reference images in conjunction with generative adversarial networks (GANs) may allow even better matches of text-illustrating images to be produced [58]. Generating images is an alternative approach to that presented in [59], where a computer method automatically selects already existing images from an album and places them in suitable contexts within a body of text. Using a GAN may result in the ability to generate realistic images instead of picture book-style drawings [60]. It may also be valuable to further validate the resulting illustrative images by re-generating text based on them and comparing it to the original sentences of the summary, for example using the methods described in [61]. Those topics are certainly worthy of further research.

Author Contributions: Conceptualization: T.H.; methodology: T.H. and J.G.; software: J.G. and T.H.; validation: J.G., T.H. and G.S.; formal analysis: T.H.; investigation, J.G. and T.H.; data curation: J.G.; writing—original draft preparation, J.G. and T.H.; writing—review and editing: J.G., T.H. and G.S.; visualization: T.H. and J.G.; funding acquisition, T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded under the Pedagogical University of Krakow statutory research grant, which was funded by subsidies for science granted by the Polish Ministry of Science and Higher Education.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Both source codes and data are available for download from the online repository https://github.com/JusMia/TIPS, accessed on 27 September 2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Long, S.; He, X.; Yao, C. Scene Text Detection and Recognition: The Deep Learning Era. *Int. J. Comput. Vis.* **2021**, *129*, 161–184. [CrossRef]
2. Yan, S.; Yu, L.; Xie, Y. Discrete-continuous Action Space Policy Gradient-based Attention for Image-Text Matching. *arXiv* **2021**, arXiv:2104.10406.
3. Hu, W.; Dang, A.; Tan, Y. *A Survey of State-of-the-Art Short Text Matching Algorithms*; Springer Nature: Singapore, 2019; pp. 211–219._22. [CrossRef]
4. Li, K.; Zhang, Y.; Li, K.; Li, Y.; Fu, Y. Visual Semantic Reasoning for Image-Text Matching. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 4653–4661. [CrossRef]
5. Jogin, M.; Mohana, M.; Madhulika, M.; Divya, G.; Meghana, R.; Apoorva, S. *Feature Extraction Using Convolution Neural Networks (CNN) and Deep Learning*; IEEE: Piscataway, NJ, USA, 2018; pp. 2319–2323. [CrossRef]
6. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019** , arXiv:1905.11946.
7. Zoph, B.; Ghiasi, G.; Lin, T.Y.; Cui, Y.; Liu, H.; Cubuk, E.; Le, Q. Rethinking Pre-training and Self-training. *arXiv* **2020**, arXiv:2006.06882.
8. Chawla, P.; Jandial, S.; Badjatiya, P.; Chopra, A.; Sarkar, M.; Krishnamurthy, B. *Leveraging Style and Content Features for Text Conditioned Image Retrieval*; IEEE: Piscataway, NJ, USA, 2021; pp. 3973–3977. [CrossRef]
9. Conde, M.; Turgutlu, K. *CLIP-Art: Contrastive Pre-training for Fine-Grained Art Classification*; IEEE: Piscataway, NJ, USA, 2021; pp. 3951–3955. [CrossRef]
10. Zhou, Y.; Wang, M.; Liu, D.; Hu, Z.; Zhang, H. *More Grounded Image Captioning by Distilling Image-Text Matching Model*; IEEE: Piscataway, NJ, USA, 2020; pp. 4776–4785. [CrossRef]
11. Joshi, D.; Wang, J. The Story Picturing Engine—A system for automatic text illustration. *TOMCCAP* **2006**, *2*, 68–89. [CrossRef]
12. Huang, F.; Zhang, X.; Li, Z.; Zhao, Z. Bi-Directional Spatial-Semantic Attention Networks for Image-Text Matching. *IEEE Trans. Image Process.* **2018**, *28*, 2008–2020. [CrossRef]
13. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. *arXiv* **2019**, arXiv:1908.10084.
14. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
15. Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; Bordes, A. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. *arXiv* **2017**, arXiv:1705.02364.
16. Cer, D.; Yang, Y.; Kong, S.Y.; Hua, N.; Limtiaco, N.; John, R.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; et al. Universal Sentence Encoder. *arXiv* **2018**, arXiv:1803.11175.
17. Conneau, A.; Kiela, D. SentEval: An Evaluation Toolkit for Universal Sentence Representations. *arXiv* **2018**, arXiv:1705.02364.
18. Devillers, B.; Bielawski, R.; Choski, B.; VanRullen, R. Does language help generalization in vision models? *arXiv* **2021**, arXiv:2104.08313v3 .
19. Zhu, J.; Li, H.; Liu, T.; Zhou, Y.; Zhang, J.; Zong, C. MSMO: Multimodal Summarization with Multimodal Output. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 4154–4164. [CrossRef]
20. Rudinac, S.; Chua, T.S.; Díaz Ferreyra, N.; Friedland, G.; Gornostaja, T.; Huet, B.; Kaptein, R.; Lindén, K.; Moens, M.F.; Peltonen, J.; et al. *Rethinking Summarization and Storytelling for Modern Social Multimedia*; Springer: Cham, Switzerland, 2018; pp. 632–644. [CrossRef]
21. Baltrušaitis, T.; Ahuja, C.; Morency, L.P. Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 423–443. [CrossRef]
22. Guo, W.; Wang, J.; Wang, S. Deep Multimodal Representation Learning: A Survey. *IEEE Access* **2019**, *7*, 63373–63394. [CrossRef]
23. Gao, J.; Li, P.; Chen, Z.; Zhang, J. A Survey on Deep Learning for Multimodal Data Fusion. *Neural Comput.* **2020**, *32*, 829–864. [CrossRef] [PubMed]
24. Ramachandram, D.; Taylor, G.W. Deep Multimodal Learning: A Survey on Recent Advances and Trends. *IEEE Signal Process. Mag.* **2017**, *34*, 96–108. [CrossRef]
25. Huddar, M.; Sannakki, S.; Rajpurohit, V. A Survey of Computational Approaches and Challenges in Multimodal Sentiment Analysis. *Int. J. Comput. Sci. Eng.* **2019**, *7*, 876–883. [CrossRef]

26. Soleymani, M.; Garcia, D.; Jou, B.; Schuller, B.; Chang, S.F.; Pantic, M. A survey of multimodal sentiment analysis. *Image Vis. Comput.* **2017**, *65*, 3–14.
27. Cai, Q.; Wang, H.; Li, Z.; Liu, X. A Survey on Multimodal Data-Driven Smart Healthcare Systems: Approaches and Applications. *IEEE Access* **2019**, *7*, 133583–133599. [CrossRef]
28. Liu, Y. Fine-tune BERT for Extractive Summarization. *arXiv* **2019**, arXiv:1903.10318.
29. Cheng, J.; Lapata, M. Neural Summarization by Extracting Sentences and Words. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 484–494. [CrossRef]
30. Narayan, S.; Cohen, S.B.; Lapata, M. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 1747–1759. [CrossRef]
31. Zhou, Q.; Yang, N.; Wei, F.; Huang, S.; Zhou, M.; Zhao, T. Neural Document Summarization by Jointly Learning to Score and Select Sentences. *arXiv* **2018**, arXiv:1807.02305.
32. Ba, J.; Kiros, J.; Hinton, G. Layer Normalization. *arXiv* **2016**, arXiv:1607.06450.
33. Celikyilmaz, A.; Bosselut, A.; He, X.; Choi, Y. Deep Communicating Agents for Abstractive Summarization. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 1662–1675. [CrossRef]
34. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the KDD04: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 168–177. [CrossRef]
35. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. *arXiv* **2015**, arXiv:1503.03832.
36. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
37. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
39. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June, doi:10.1109/CVPR.2016.308. [CrossRef]
40. Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
41. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3156–3164. [CrossRef]
42. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *arXiv* **2015**, arXiv:1502.03044.
43. Tanti, M.; Gatt, A.; Camilleri, K. Where to put the Image in an Image Caption Generator. *Nat. Lang. Eng.* **2017**, *24*, 467–489. [CrossRef]
44. Amirian, S.; Rasheed, K.; Taha, T.; Arabnia, H. Automatic Image and Video Caption Generation With Deep Learning: A Concise Review and Algorithmic Overlap. *IEEE Access* **2020**, *8*, 218386–218400. [CrossRef]
45. Bernardi, R.; Cakici, R.; Elliott, D.; Erdem, A.; Erdem, E.; Ikizler, N.; Keller, F.; Muscat, A.; Plank, B. Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures. *J. Artif. Intell. Res.* **2016**, *55*. [CrossRef]
46. Wang, J.; Dong, Y. Measurement of Text Similarity: A Survey. *Information* **2020**, *11*, 421. [CrossRef]
47. Radford, A.; Kim, J.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. *arXiv* **2021**, arXiv:2103.00020.
48. Sohn, K. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *Advances in Neural Information Processing Systems*; Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2016; Volume 29.
49. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
50. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
51. Miller, D. Leveraging BERT for Extractive Text Summarization on Lectures. *arXiv* **2019**, arXiv:1906.04165.
52. Qiu, Y.; Jin, Y. Engineering Document Summarization Using Sentence Representations Generated by Bidirectional Language Model. In Proceedings of the ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual, 17–19 August 2021. [CrossRef]
53. To, H.; Nguyen, K.; Nguyen, N.; Nguyen, A. Monolingual versus Multilingual BERTology for Vietnamese Extractive Multi-Document Summarization. *arXiv* **2021**, arXiv:2108.13741.

54. Srikanth, A.; Umasankar, A.S.; Thanu, S.; Nirmala, S.J. Extractive Text Summarization using Dynamic Clustering and Co-Reference on BERT. In Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS), Virtual, 14–16 October 2020; pp. 1–5. [CrossRef]
55. Marcelino, G.; Semedo, D.; Mourão, A.; Blasi, S.; Mrak, M.; Magalhães, J. A Benchmark of Visual Storytelling in Social Media. *arXiv* **2019**, arXiv:1908.03505.
56. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Volume 8693. [CrossRef]
57. Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; et al. The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *Int. J. Comput. Vis.* **2020**, *128*, 1956–1981. [CrossRef]
58. Li, B.; Qi, X.; Lukasiewicz, T.; Torr, P. ManiGAN: Text-Guided Image Manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 7877–7886. [CrossRef]
59. Nag Chowdhury, S.; Cheng, W.; de Melo, G.; Razniewski, S.; Weikum, G. Illustrate Your Story: Enriching Text with Images. In Proceedings of the WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 849–852. [CrossRef]
60. Qi, X.; Song, R.; Wang, C.; Zhou, J.; Sakai, T. Composing a Picture Book by Automatic Story Understanding and Visualization. In Proceedings of the Second Workshop on Storytelling, Florence, Italy, 1 August 2019; pp. 1–10. [CrossRef]
61. Hu, J.; Cheng, Y.; Gan, Z.; Liu, J.; Gao, J.; Neubig, G. What Makes A Good Story? Designing Composite Rewards for Visual Storytelling. *arXiv* **2019**, arXiv:1909.05316.